An Overture Overview

Bill Henshaw

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA

Twelfth DOE ACTS Workshop,
Berkeley California, 2011

Downloading Overture and the CG (Composite Grid) suite of PDE solvers.

Overture and CG are freely available from the web:

www.llnl.gov/CASC/Overture

Acknowledgments.

Supported by

Department of Energy, Office of Science

ASCR Applied Math Program

LLNL: Laboratory Directed Research and Development (LDRD) program

Current Overture developers

Kyle Chand

Bill Henshaw

Major Contributors

Don Schwendeman (RPI),

Jeff Banks (LLNL).

Overture: a toolkit for solving partial differential equations (PDEs) on overlapping grids.

Top three reasons for using Overture:

1. You need to efficiently solve a PDE on a complex geometry.
2. You need to solve a PDE on a moving geometry.
3. You need to generate an overlapping grid.

You can

- write your own PDE solver using the capabilities provided by Overture.
- use (or change) an existing PDE solver from the CG suite.

Overture: a toolkit for solving partial differential equations (PDEs) on overlapping grids.

Top three reasons for using Overture:

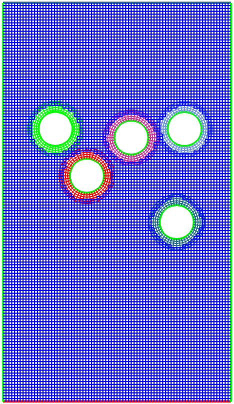
1. You need to efficiently solve a PDE on a complex geometry.
2. You need to solve a PDE on a moving geometry.
3. You need to generate an overlapping grid.

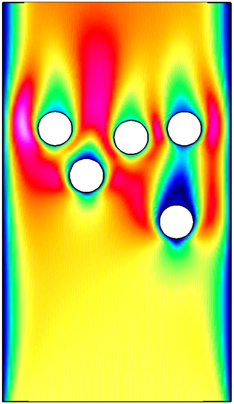
You can

- write your own PDE solver using the capabilities provided by Overture.
- use (or change) an existing PDE solver from the CG suite.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

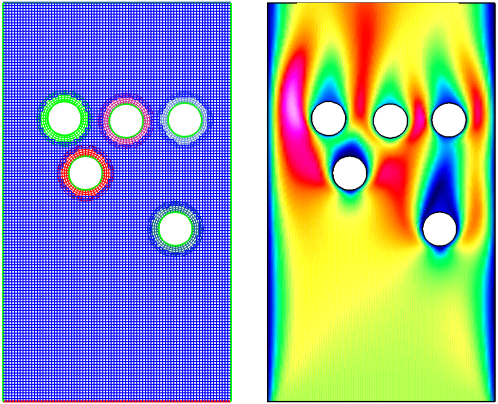




- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

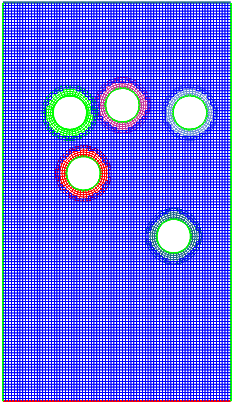
Overlapping grid: a set of structured grids that overlap.

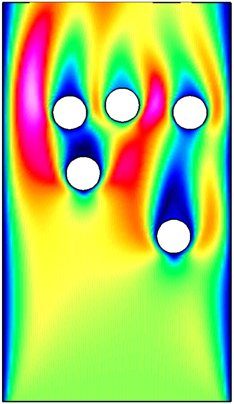


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

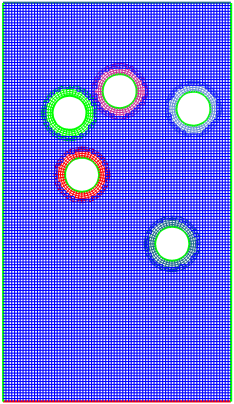


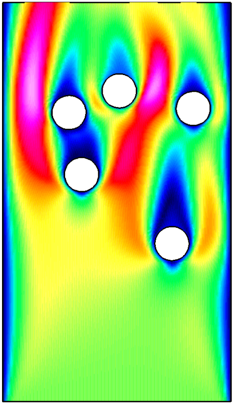


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

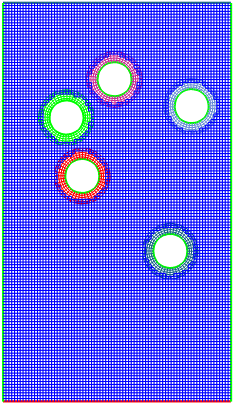


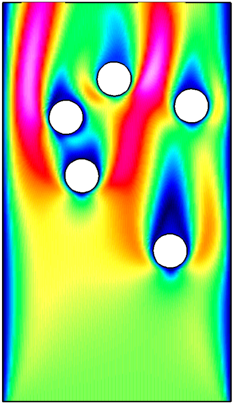


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.

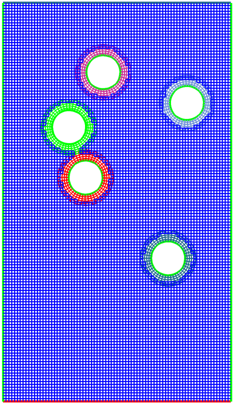


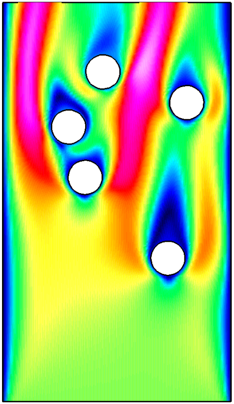


- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

What are overlapping grids and why are they useful?

Overlapping grid: a set of structured grids that overlap.





- Overlapping grids can be rapidly generated as bodies move.
- High quality grids under large displacements.
- Cartesian grids for efficiency.
- Efficient for high-order accurate methods.

Key Features of Overture

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

Key Features of Overture

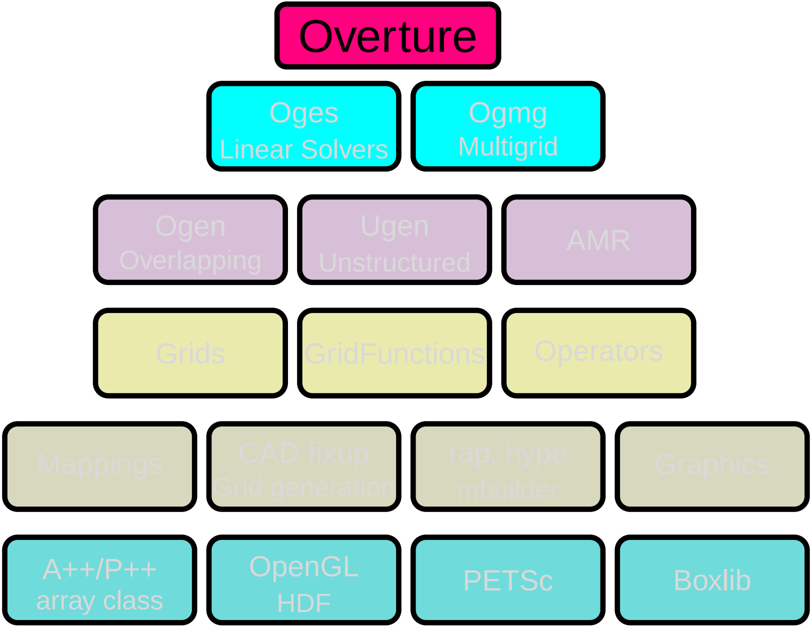
- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

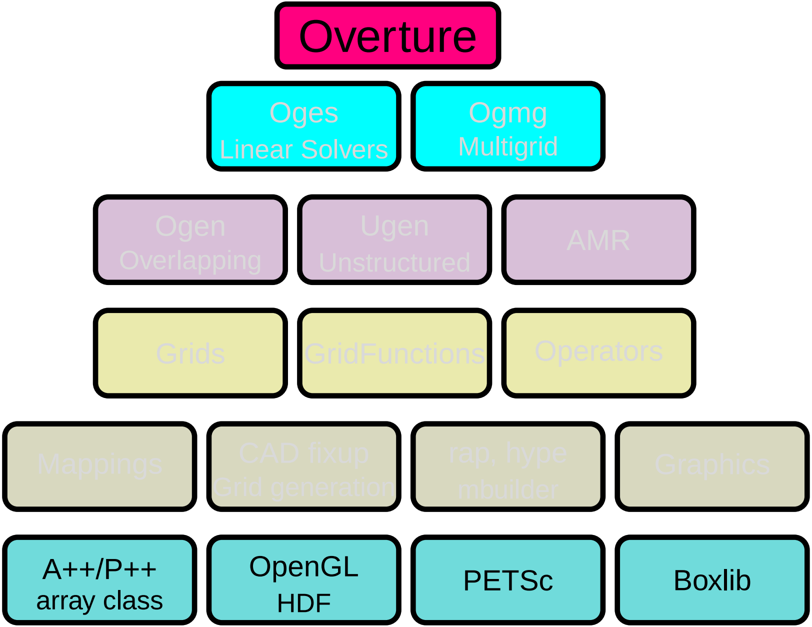
Key Features of Overture

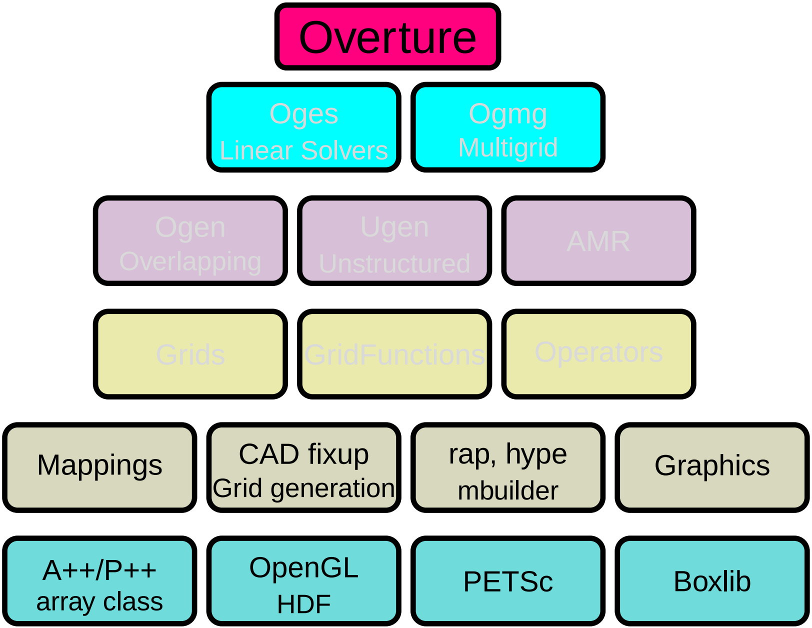
- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).

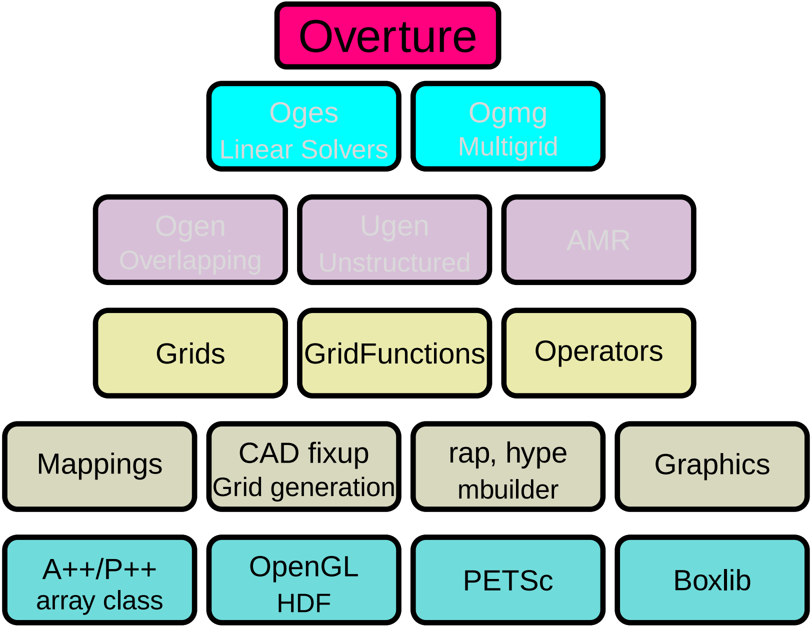
Key Features of Overture

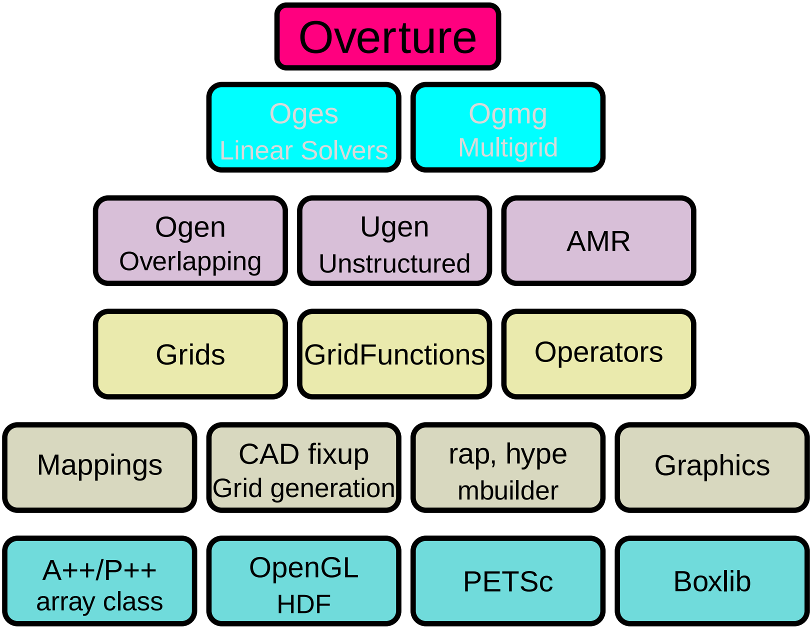
- high level C++ interface for rapid prototyping of PDE solvers.
- built upon optimized C and fortran kernels.
- library of finite-difference operators: conservative and non-conservative, 2nd, 4th, 6th and 8th order accurate approximations.
- support for moving grids.
- support for block structured adaptive mesh refinement (AMR).
- extensive grid generation capabilities.
- CAD fixup tools (for CAD from IGES files).
- interactive graphics and data base support (HDF).


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib

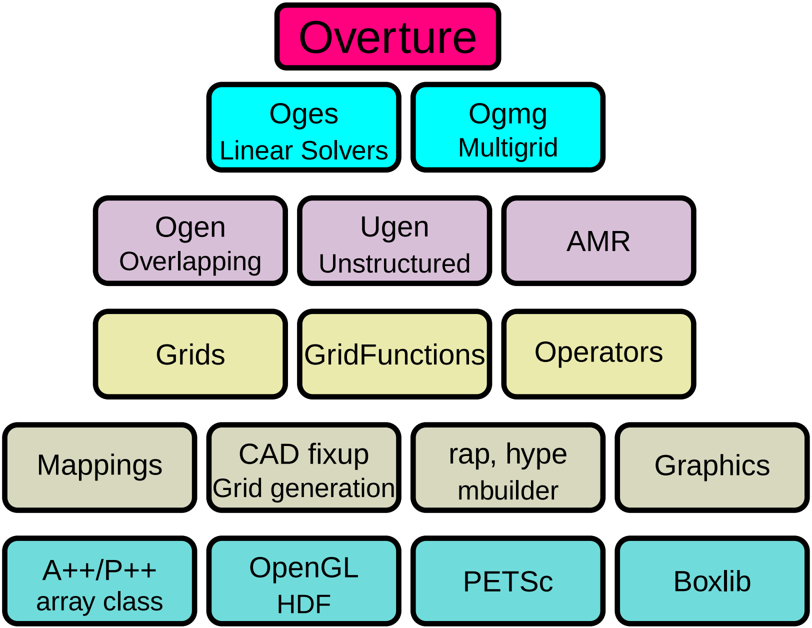

Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib



Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib


Overture
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid generation
rap, hype
mbuilder
Graphics
A++/P++
array class
OpenGL
HDF
PETSc
Boxlib

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

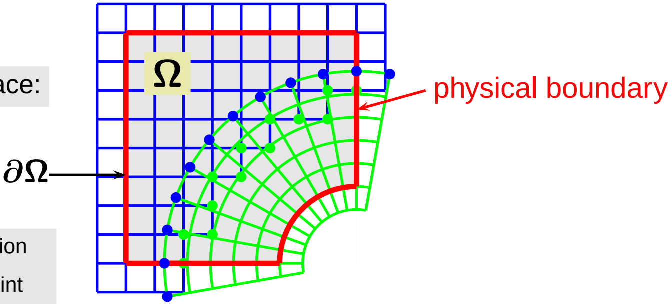
CG: Composite Grid PDE solvers built with Overture

Different PDE solvers in the CG suite:

- cgad: advection diffusion equations.
- cgins: incompressible Navier-Stokes with heat transfer.
- cgcns: compressible Navier-Stokes, reactive Euler equations.
- cgmp: multi-physics solver (e.g. conjugate heat transfer).
- cgmx: time domain Maxwell's equations solver.
- cgsm: elastic wave equation (linear elasticity).

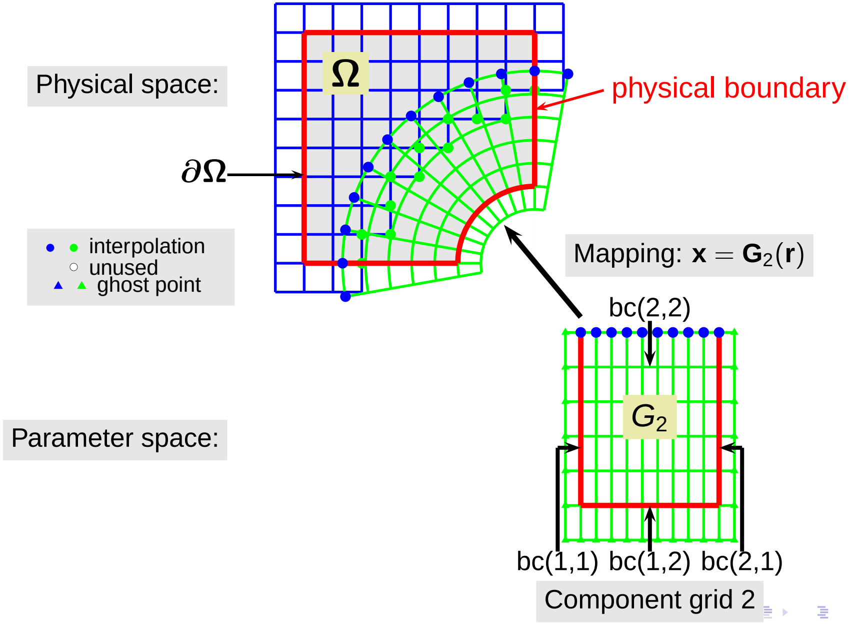
Components of an Overlapping Grid

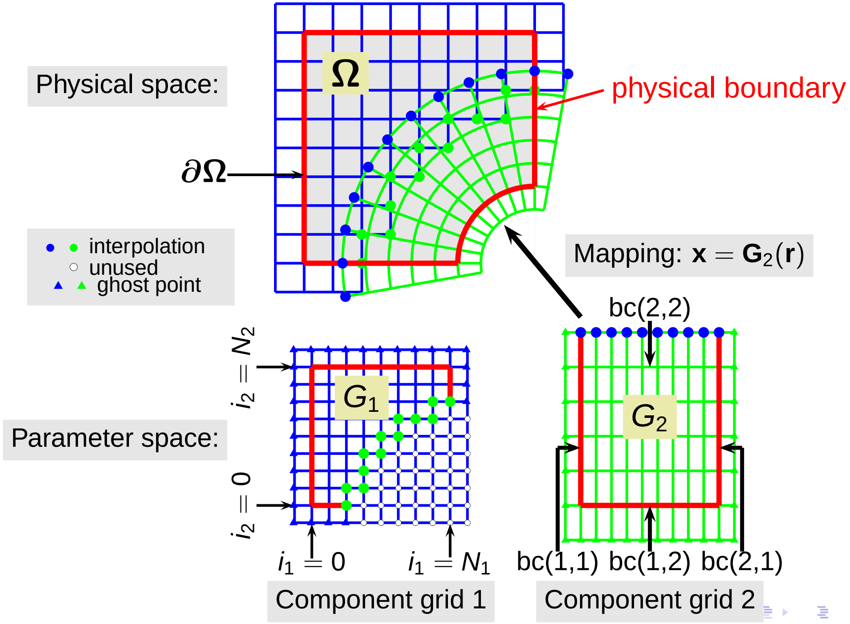
Physical space:



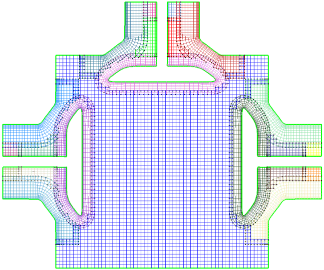
● ● interpolation
○ unused
▲ ▲ ghost point

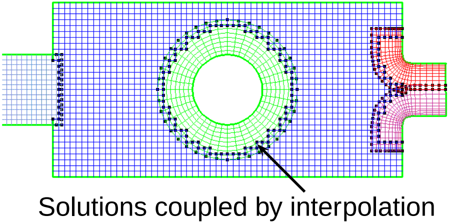
Components of an Overlapping Grid

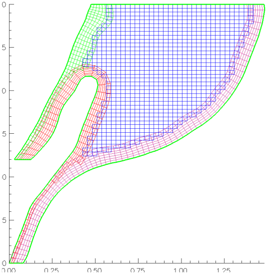



Physical space:
Ω
∂Ω
physical boundary
interpolation
unused
ghost point
Mapping: $\mathbf{x} = \mathbf{G}_2(\mathbf{r})$
Parameter space:
G_1
$i_2 = N_2$
$i_2 = 0$
$i_1 = 0$
$i_1 = N_1$
G_2
bc(2,2)
bc(1,1) bc(1,2) bc(2,1)
Component grid 1
Component grid 2

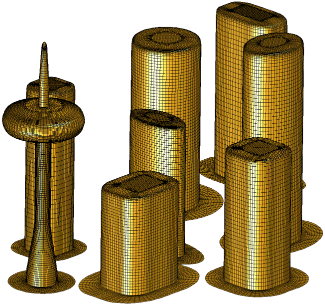
Ogen can be used to build 2D overlapping grids:

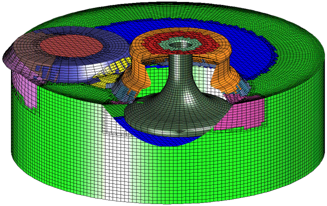


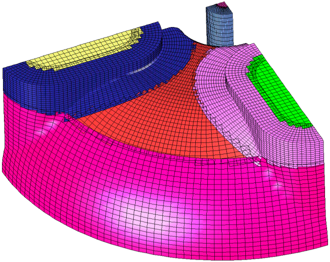


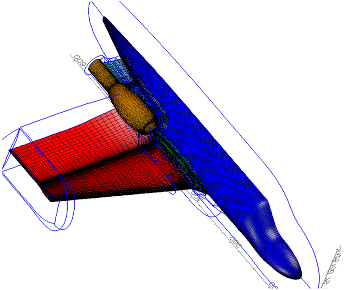


Ogen can be used to build 3D overlapping grids:









Overture supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg; // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg); // create a grid function
u=1.;
CompositeGridOperators op(cg); // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
   u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
   t+=dt;
   u.interpolate();
   u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
   u.finishBoundaryConditions();
}
```

Overture supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg; // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg); // create a grid function
u=1.;
CompositeGridOperators op(cg); // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
  u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
  t+=dt;
  u.interpolate();
  u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
  u.finishBoundaryConditions();
}
```

Overture supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg; // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg); // create a grid function
u=1.;
CompositeGridOperators op(cg); // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
   u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
   t+=dt;
   u.interpolate();
   u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
   u.finishBoundaryConditions();
}
```

Overture supports a high-level C++ interface

But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg; // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg); // create a grid function
u=1.;
CompositeGridOperators op(cg); // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
   u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
   t+=dt;
   u.interpolate();
   u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
   u.finishBoundaryConditions();
}
```

Overture supports a high-level C++ interface

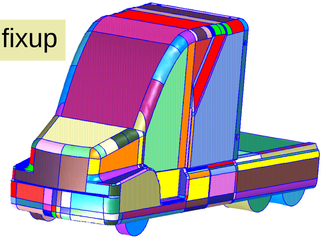
But is built upon mainly Fortran kernels.

Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg; // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg); // create a grid function
u=1.;
CompositeGridOperators op(cg); // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
   u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
   t+=dt;
   u.interpolate();
   u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
   u.finishBoundaryConditions();
}
```

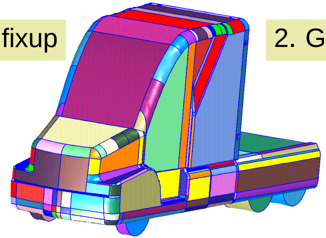
From CAD to Mesh to Solution with Overture

1. Cad fixup

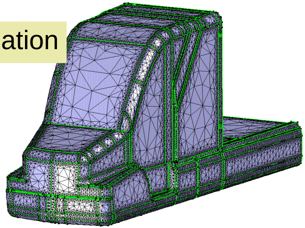


From CAD to Mesh to Solution with Overture

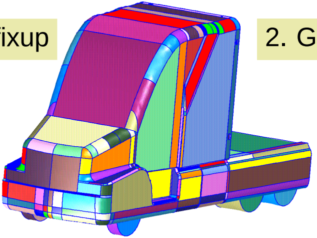
1. Cad fixup



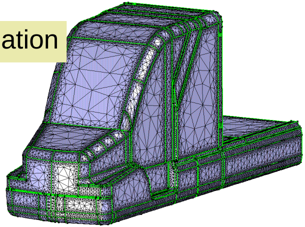
2. Global triangulation



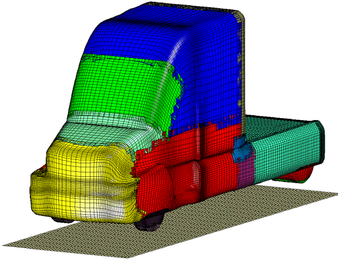
1. Cad fixup



2. Global triangulation

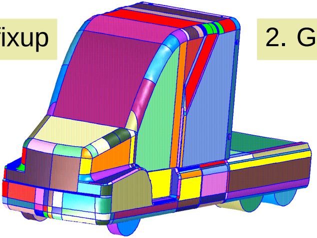


3. Overlapping grid

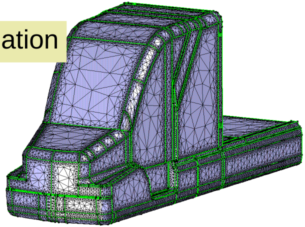


From CAD to Mesh to Solution with Overture

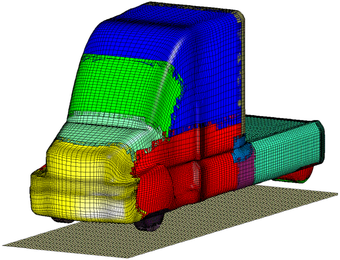
1. Cad fixup



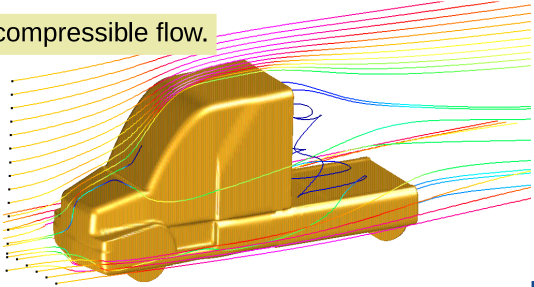
2. Global triangulation



3. Overlapping grid



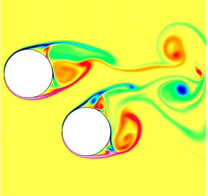
4. Incompressible flow.

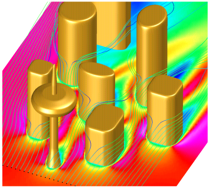


Overture is used by research groups worldwide

- Blood flow in veins with blood clot filters. (Mike Singer, LLNL).
- Pitching airfoils and micro-air vehicles (Yongsheng Lian, U. of Louisville)
- Relativistic hydrodynamics and Einstein field equations (Philip Blakely, Nikos Nikiforakis, U. Cambridge).
- Compressible flow/ice-formation (Graeme Leese, U. Cambridge).
- Tear films and droplets (Rich Braun U. Delaware, Kara Maki UMN).
- High-order accurate subsonic/transonic aero-acoustics (Phillipe Lafon, CNRS, EDF, France).
- Low Reynolds flow for pitching airfoils (D. Chandar, R. Yapalparvi, M. Damodaran, NTU, Singapore).
- Incompressible flow in pumps (J.P. Potanza, Shell Oil, Houston).
- High-order accurate, compact Hermite-Taylor schemes (Tom Hagstrom, SMU, Dallas).

Cgins: incompressible Navier-Stokes solver.



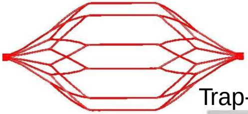


- 2nd-order and 4th-order accurate (DNS).
- support for moving rigid-bodies (not parallel yet).
- heat transfer (Boussinesq approximation).
- semi-implicit (time accurate), pseudo steady-state (efficient line solver), full implicit.

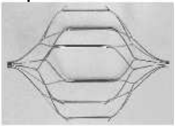
• WDH., *A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids*, J. Comput. Phys, **113**, no. 1, (1994) 13–25.

Flow past a blood-clot filter using cgins

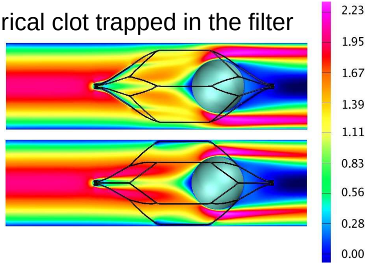
Overlapping grid for the filter



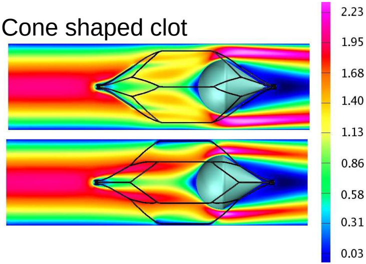
Trap-ease wire filter



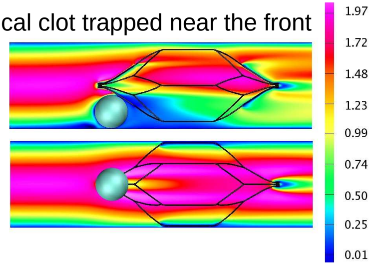
Spherical clot trapped in the filter



Cone shaped clot

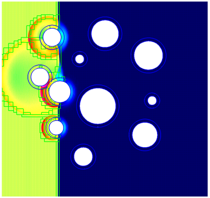


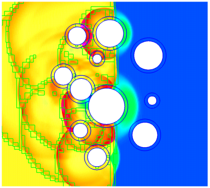
Spherical clot trapped near the front



M.A. Singer, WDH, S.L. Wang, *Computational Modeling of Blood Flow in the Trapease Inferior Vena Cava Filter*, Journal of Vascular and Interventional Radiology, **20**, 2009.

Cgcns: compressible N-S and reactive-Euler.

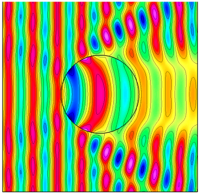


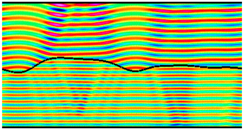


- reactive and non-reactive Euler equations, Don Schwendeman (RPI).
- compressible Navier-Stokes.
- multi-fluid formulation, Jeff Banks (LLNL).
- adaptive mesh refinement and moving grids.

• WDH., D. W. Schwendeman, *Parallel Computation of Three-Dimensional Flows using Overlapping Grids with Adaptive Mesh Refinement*, J. Comp. Phys. **227** (2008).
• WDH., DWS, *Moving Overlapping Grids with Adaptive Mesh Refinement for High-Speed Reactive and Nonreactive Flow*, J. Comp. Phys. **216** (2005).
• WDH., DWS, An adaptive numerical scheme for high-speed reactive flow on overlapping grids, J. Comp. Phys. **191** (2003).

Cgmx: electromagnetics solver.





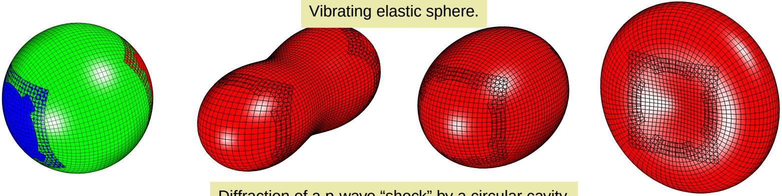
- fourth-order accurate, 2D, 3D.
- Efficient time-stepping with the modified-equation approach
- High-order accurate symmetric difference approximations.
- High-order-accurate *centered* boundary and interface conditions.

• WDH., *A High-Order Accurate Parallel Solver for Maxwell's Equations on Overlapping Grids*, SIAM J. Scientific Computing, **28**, no. 5, (2006).

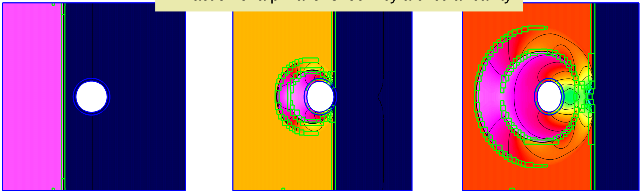
Cgsm: solve the elastic wave equation.

- linear elasticity on overlapping grids, with adaptive mesh refinement,
- conservative finite difference scheme for the second-order system,
- upwind Godunov scheme for the first-order-system.

Vibrating elastic sphere.



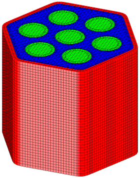
Diffraction of a p-wave “shock” by a circular cavity.

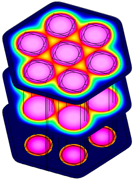


- D. Appelö, J.W. Banks , WDH, D.W. Schwendeman, *Numerical Methods for Solid Mechanics on Overlapping Grids: Linear Elasticity*, LLNL-JRNL-422223, submitted.

Cgmp: a multi-domain multi-physics solver.

Conjugate heat transfer: coupling incompressible flow to heat conduction in solids.

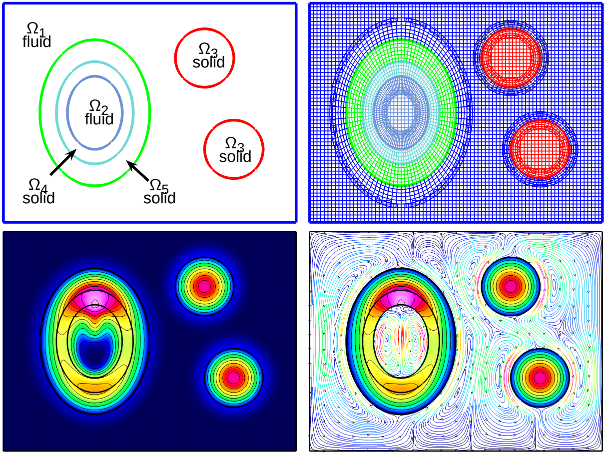




- overlapping grids for each fluid or solid domain,
- a partitioned solution algorithm (separate physics solvers in each sub-domain),
- (cgins) incompressible Navier-Stokes equations (with Boussinesq approximation) for fluid domains,
- (cgad) heat equation for solid domains,
- a key issue is interface coupling.

• WDH., K. K. Chand, *A Composite Grid Solver for Conjugate Heat Transfer in Fluid-Structure Systems*, J. Comput. Phys, 2009.

The multi-domain composite grid approach



The fluid and solid sub-domains, overlapping grids and solution (temperature and streamlines) to a CHT problem. Solvers: cgins (fluid sub-domains), cgad (solid sub-domains), cgmp (coupled problem).

Summary.

- Overture: a toolkit for solving PDEs on overlapping grids.
- CG : a suite of PDE solvers for overlapping grids.

www.llnl.gov/CASC/Overture